1.0 Introduction

Lessons Learned Note *click here* for Lessons Learned *that may apply* to the requirements contained in this LIR.

1.1 Background

Quality assurance is to be integrated into the Laboratory's software engineering processes by implementation of the requirements contained in this LIR. To ensure the development of quality software products, these requirements establish the required rigor for the software life-cycle processes and technical work products. To protect the integrity of the software, the requirements also establish the processes for software management and encompass all aspects of the software life cycle from initial planning through production and maintenance. The processes required in this LIR are tailored from the key process areas associated with Levels 2 and 3 of the Software Engineering Institute's Capability Maturity Model (CMM).

This LIR requires the following:

1. Implementation of industry best practices throughout the software development life cycle.

2. Software management processes that are tailored to the work and its associated hazards as well as the level of effort, complexity and degree of impact of the software product.

This LIR supports the expectations contained in LPR 308-00-00, Integrating Quality Management and the good business practices of project management, configuration management, records management and security.

This document becomes effective on the issue date. The Implementation Plan (Attachment C of this LIR) must be completed within four months of the original issue date.

1.2 In this Document

Table of Contents

1.2 In this Document

Table of Contents

2.0 Purpose

The purpose of the requirements contained in this LIR shall be to promote the development of reliable, cost-effective, software products while making efficient use of Laboratory resources through the application of a risk-based graded approach resulting in the implementation of specific software process rigor.

3.0 Scope/Applicability

The requirements contained in this LIR shall apply to all Laboratory-related software that has been or currently is being developed, modified or acquired by the Laboratory and performs calculations or data transformations that will be relied upon by more than one person, or affects the health/safety, security, environmental or programmatic impact areas, as described in Attachment A of this LIR.

This scope includes Commercial or Government Off-The-Shelf software (OTS), scripts, macros, legacy software and software currently under development or planned for development. Refer to the definition of software included in section 4.2 below.

This LIR, through its companion Implementation Plan (see Attachment C), requires each division at the Laboratory to identify, grade and control software under the domain of the division. The division implementation plans shall apply to all software as defined in this LIR and shall provide division-specific approaches for the management of software projects, products, scripts, macros

and other large or small codes in accordance with the risks imposed by their use. Division implementation plans shall also provide adequate coverage for intended use of OTS software, where processes, procedures and documentation address only the use for which the OTS software is intended, not all functions available in the OTS software.

Guidance Notes:

1. Software intended for casual sharing or casual use, i.e. not deliverable as a product, may be exempted by the division from the requirements of this LIR until it is intended for use in support of programmatic or operational purposes. For example, software developed in support of research efforts that do not have the goal of delivering a software product may be exempted by the division.

2. It is impractical to grade and control small codes, scripts and macros on an individual basis. Division implementation plans may address approaches for consolidating small codes into libraries or larger projects as applicable.

3. It is impractical to develop a complete requirements specification and test plan for all functions and features of OTS software, for example Microsoft Excel. It *is* practical to develop a requirements specification and test plan for spreadsheets developed with Microsoft Excel and which are graded as Category 1 or Category 2 software as described in Attachment A of this LIR.

4.0 Definitions

4.1 Acronyms

AD—Associate Director

CIO—Chief Information Officer

LIR—Laboratory Implementation Requirements

OTS—Commercial or Government-developed Off-The-Shelf software

SPMP—Software Project Management Plan

SQA—Software Quality Assurance

SRS—Software Requirements Specification

SVVP—Software Verification and Validation Plan

4.2 Terms

category 1 software—A software project or product that has been determined as having catastrophic or critical impact in one or more of the identified impact areas as defined in Attachment A, Section 4 of this LIR.

category 2 software—A software project or product that has been determined as having moderate to minimal impact in any of the identified impact areas as defined in Attachment A, Section 4 of this LIR and is not category 1 software.

category 3 software—A software project or product that has been determined as having minimal to negligible impact in all of the identified impact areas as

defined in Attachment A, Section 4 of this LIR.

graded approach—A process implemented to determine the depth of detail required and the magnitude of resources expended for a particular management element to be tailored commensurate with the relative importance of the software project to the work, safety, health, environmental compliance, safeguards and security, programmatic importance, magnitude of the hazard, and financial impact.

probable—Likely to occur or reasonably expected.

possible—Slight chance of occurrence or highly unlikely to occur.

not credible—Effectively zero probability of occurrence.

software—A computer program or suite of programs, procedures or routines, having material value and usefulness that enable a computer system to process data.

Material value	Automates Laboratory operations and/or is delivered for use or evaluation by one or more Laboratory customers.
Usefulness	Is actively used today.
Enables	Provides tools for and allows access to.
Process	Performing data entry, data storage, data retrieval, calculations, or any other data manipulation.
Data	Information organized for analysis or computation.

Subject Matter Experts (SMEs)—For the purpose of this document, SMEs are personnel knowledgeable in software development, software engineering or in the intended application domains for the software.

Software Project or Product - Development and/or acquisition of software for the intended use at LANL in the accomplishment of its mission or for delivery to Laboratory customers or collaborators.

5.0 Precautions and Limitations

None

6.0 Implementation Requirements

6.1 Personnel/ Organizations

Who Person/Organization	Shall
Lab Director	• Ensure that an institutional software quality management program is initiated and sustained.
CIO Office	• Be responsible for the development and execution of the institutional software quality management implementation plan. • Be responsible for oversight of the institutional software quality management program and the requirements contained in this document.
Associate Directors, Division Leaders and Group Leaders	• Ensure that software projects or products under their purview are graded according to the provisions described in Attachment A of this document. • Ensure that the processes associated with Category 1 and/or Category 2 software projects or products are implemented according to the provisions of this document. (See Section 6.2.2.) • Establish mechanisms to monitor software projects or products within their purview. • Ensure that Laboratory subcontractors performing software development implement the requirements of this LIR as part of their contract.
Software Project Leaders or lead software developers	• Implement processes that apply to Category 1 or Category 2 software projects or products in accordance with the requirements contained in this document. (See Section 6.2.2.) • Ensure documentation associated with the requirements contained in this document is maintained in accordance with organizational records management requirements.
Software Developers	• Perform and document their work in accordance with the requirements contained in this document.
End Users (Laboratory workers)	• Provide feedback concerning functional accuracy, errors or usability of the software to the software developer or development team.

6.2 Required Processes

6.2.1 Software Grading Documentation

For all software projects or products subject to the requirements of this LIR, the worksheet as shown in Attachment B must be completed and filed with the software developer's group and/or program office, as required by the division.

6.2.2 Software Management Processes

Category 1 Software

A Category 1 software project or product is one that has been determined as having catastrophic or critical impact in one or more of the identified impact areas. To mitigate risk, full implementation of software process rigor in project planning, implementation and deployment shall be required.

For Category 1 software, the processes in the following table shall be documented, approved by the responsible line manager (division/directorate) and implemented and monitored.

Process	Shall Include
Formal Training and Qualification Process	A set of activities to define qualifications for and provide training to software project managers and developers. This process further includes the development of training and qualification requirements for software end users, commensurate with the risks associated with the use of the software.
Software Configuration Management Process	A process that effectively controls the coordination and implementation of changes to software components. Activities include defect and enhancement identification, prioritization and selection; document and code version tracking and control; and software installation tracking. **Guidance Note:** This process is typically documented in a Software Configuration Management Plan (SCMP).
Software Product Engineering Process	A set of software development activities or stages that function together to guide the development and maintenance of a software product. **Guidance Note:** The generally accepted stages for software development are Planning, Requirements, Design, Development, Integration & Test, Installation & Acceptance, and Maintenance. Each of these stages is finite in scope, requires a specific set of inputs, and produces a specific set of deliverables.
Software Project Management Process	The technical and management approach to be followed for a software project.

Process	Shall Include
	The process encompasses the scope of work, roles and responsibilities, the methods to be used, the project work breakdown structure and initial schedule, as well as a list of deliverables and other key events required for the project to be considered a success. **Guidance Note:** This process is typically documented in a Software Project Management Plan (SPMP) and associated training and guidance.
Software Project Tracking and Oversight Process	A set of activities to monitor and evaluate a specific set of performance metrics associated with the success of the project and the quality of the software under development. **Guidance Note:** Typical metrics tracked include schedule and budget compliance, level of effort, size of the software product, defects identified and enhancements requested. These metrics are typically used to initiate corrective actions when project objectives are not being met.
Software Quality Assurance Process	A set of activities designed to evaluate the process by which software products are developed. A formal process for evaluating the work products produced during the software development life cycle. **Guidance Note:** This may be accomplished through specific reviews and assessment by peers and independent quality assurance reviewers. This process is typically documented in a Software Quality Assurance Plan (SQAP).
Software Requirements Management Process	A process for identifying and controlling conditions or capabilities needed by the customer to solve a problem or achieve an objective. Software requirements are conditions or capabilities that must be met or possessed by the developed software before it will be accepted by the customer.

Process	Shall Include
	Guidance Note: This process is typically documented as part of a Software Verification and Validation Plan (SVVP), with the results of the process documented in a Software Requirements Specification (SRS).
Software Risk Management Process	A process of problem analysis that is used to identify, analyze, prioritize, monitor and control risks. The level of effort and documentation associated with risk management is commensurate with the level of initial and mitigated risk associated with the development and use of the software. **Guidance Note:** This process is typically documented as a feasibility and risk analysis and may be a separate document or incorporated into the SPMP.
Software Verification and Validation Process	The processes to be used to verify traceability and validate correct functionality of the software during the software life cycle. **Guidance Note:** Verification is the process of determining whether or not the products of a given stage of the software development life cycle fulfill the requirements established during the previous stage. Validation is the process of evaluating software during and at the end of the software development process to ensure compliance with software requirements. This process is typically documented in a Software Verification and Validation Plan (SVVP), which describes the entire software development life cycle, the activities therein and the deliverables to be produced.

Category 2 Software

A Category 2 software project or product is one that has been determined as having moderate to minimal impact in any of the identified impact areas and is not Category 1 software. To mitigate risk, a lower level of software process shall be required.

For Category 2 software, the following processes shall be documented, approved by the responsible line manager and implemented to a degree commensurate with the intended use of the software.

- Software Configuration Management Process
- Software Product Engineering Process
- Software Project Management Process
- Software Project Tracking and Oversight Process
- Software Quality Assurance Process
- Software Requirements Management Process
- Software Verification and Validation Process

Guidance Note: The level of implementation is not required to be as rigorous as that for Category 1 software projects or products and may be tailored at the project level.

Category 3 Software

A Category 3 software project or product is one that has been determined as having minimal to negligible impact in all of the impact areas. Accordingly, no additional processes shall be required beyond the grading process defined in Section 6.2.1 above.

7.0 Documentation

None

8.0 References

8.1 Document Ownership

The office of institutional coordination (OIC) for this document shall be the CIO office (7-0961).

8.2 Driver Documents

8.2.1 Contractual Commitments

10 CFR 830 Subpart A, Nuclear Safety Management Quality Assurance Program
10 CFR 820, Procedural Rules for DOE Nuclear Facilities
DOE Order 414.1, Quality Assurance

8.2.2 Programmatic Commitments

DOE/AL Quality Criteria (QC-1)
U.S. Department of Energy, Carlsbad Field Office (CBFO), Quality Assurance Program Document
TA-55 Transuranic Waste Interface Document (TWID) for Debris Waste

8.2.3 Los Alamos National Laboratory Internal Commitments

LPR 308-00-00, Integrating Quality Management

9.0 Attachments

Attachment A: Risk-Based Graded Approach for Software Quality Management
Attachment B: Risk-Based Graded Approach Worksheet for Software Management Category Determination (Form 2066 [9/03])
Attachment C: Institutional Software Quality Management Implementation Plan

Attachment A

Risk-Based Graded Approach for Software Quality Management

1. Introduction to Grading

This attachment contains the requirements that when implemented shall ensure that required controls are incorporated into software projects, depending on the risk and potential negative consequences that could be associated with the application of the software.

Software projects with greater initial risk shall require more restrictive quality assurance controls, while projects with low levels of risk shall require a correspondingly less restrictive set of controls. The goal shall be to reduce risk by implementing required software quality assurance (SQA) controls.

The graded approach requires the implementation of software quality assurance rigor based on the actual risk identified for the software project in the four impact areas: health/safety, environmental, programmatic, and security. This grading requirement shall be applied when selecting required and suggested software development and quality assurance processes and practices.

2. Scope/Applicability

The grading requirements established herein shall apply to **all** Laboratory-related software development and acquisition as described in Section 3.0 of this LIR.

Guidance Note: When applying the graded approach requirements to software projects or software acquisition contracts, effective implementation includes sound engineering and management judgment, as well as accurate technical input.

3. The Graded Approach

A graded approach shall be implemented for software projects to provide a basis of analysis, documentation, and actions taken to ensure the process requirements are applied and shall be commensurate with health and safety, environment, security, and the mission of the Laboratory.

The implementation of a graded approach shall ensure that controls are defined for achieving the protection of the worker, the public, the environment, security, and the mission impact of the Laboratory (programmatic impact). Each impact area shall be evaluated separately to ensure the level of rigor required for each is established.

The grading process shall be implemented when initiating a software project or the development of a software product. A modification to an assigned Category shall be warranted if the scope of the software project or product changes and the changes have the potential to change the impact to the public or worker health safety, the environment, Laboratory security or the Laboratory mission.

If sufficient information for determining the final grading Category of a software project or product is not initially available, a preliminary Category shall be assigned and documented. Once the required information is available, the final grading Category shall be assigned and documented.

4. Impact Areas

Each project shall be evaluated in four impact areas (Health/Safety, Environmental, Programmatic and Security) to ensure the identification of the potential for catastrophic, critical, moderate or negligible impact. For each impact area and level of impact, a likelihood of occurrence of *probable, possible* or *not credible* shall be assigned.

Guidance Note: The definitions of these impact levels are given by impact area in the worksheet in Attachment B.

The worksheet in Attachment B shall be completed to determine the Category defined by the requirements contained in section 6.2.2 of LIR 308-00-05.

Attachment B
Risk-Based Graded Approach Worksheet for Software Management Category Determination

The worksheet that appears on the following pages is also available as a fillable Acrobat form at http://isqm.lanl.gov/training/get_started/rbga_form.pdf.

Software Graded Approach Risk Evaluation Worksheet

Software Name: _____ **Date:** _____

Developer / Project Manager: _____ **Z #:** _____

Category

Software Type:

	Information System (database, web site, doc mgmt)
	Scientific Modeling & Data Acquisition Software
	Facility, Process and Instrumentation Control Software
	Modification of acquired or support software

Software Purpose:

Instructions:
For each table below, indicate *the consensus of the project sponsor as well as the development team* as to the probability of a worst-case failure (see definition below) occurring for each area of impact. Check the number in the appropriate box.

Definition: Worst-case failure

For the purposes of this exercise, a worst-case failure of the software under consideration is defined as the worst possible failure occurring at the worst possible time during actual operations. In other words, a worst-case failure is the failure or failures that could generate the worst possible impacts.

Health / Safety Impacts		Probable	Possible	Not Credible
Catastrophic	Death, severe injury/occupational illness	☐ 1	☐ 1	☐ 3
Critical	Major injury/ chronic impairment or occupational illness	☐ 1	☐ 1	☐ 3
Moderate	Minor injury/temporary impairment or occupational illness	☐ 2	☐ 2	☐ 3
Negligible	Negligible injury or occupational illness	☐ 3	☐ 3	☐ 3

Environmental Impacts		Probable	Possible	Not Credible
Catastrophic	Severe environmental harm or liability	☐ 1	☐ 1	☐ 3
Critical	Major environmental harm or liability	☐ 1	☐ 2	☐ 3
Moderate	Minor environmental harm or liability	☐ 2	☐ 3	☐ 3
Negligible	Negligible environmental harm or liability	☐ 3	☐ 3	☐ 3

Programmatic Impacts		Probable	Possible	Not Credible
Catastrophic	Failure to meet program requirements leading to cancellation of high-profile project, severe property damage and/or a serious loss of public confidence.	☐ 1	☐ 1	☐ 3
Critical	Failure to meet major program requirements leading to major project rescoping, failure of program to meet major milestones or commitments, major property damage and/or some loss in public confidence.	☐ 1	☐ 2	☐ 3
Moderate	Failure to meet some program requirements leading to minor project rescoping, minor property damage and/or minor loss in public confidence.	☐ 2	☐ 3	☐ 3
Negligible	Acceptable impact on program requirements, negligible property damage and/or no loss in public confidence.	☐ 3	☐ 3	☐ 3

Security Impacts		Probable	Possible	Not Credible
Catastrophic	Classified material breach.	☐ 1	☐ 1	☐ 3
Critical	Breach involving unclassified, but sensitive subjects related to limited distribution, high intellectual property, or high liability issues.	☐ 1	☐ 2	☐ 3
Moderate	Breach involving unclassified, but non-sensitive subjects related to low intellectual property or some liability issues.	☐ 2	☐ 3	☐ 3
Negligible	Breach of unclassified, non-sensitive information.	☐ 3	☐ 3	☐ 3

Conclusion: If you checked a **1** in any box, you have a **Category 1** project. If you checked a **2** in any box, you have a **Category 2** project. Otherwise, you have a **Category 3** project. If you have a Category 1 or 2 project, proceed to guidance for your project type and category.
Action: File this form with your group or program management office.

Glossary of Terms

Acceptable Risk:

The level at which the benefits of the work outweigh the risk posed, and the risk is reduced to a level as low as is reasonably achievable.

Category 1 Software Project or product:

A Category 1 software project or product carries catastrophic or critical impact in one or more of the identified assessment areas. To mitigate risk, full implementation of software quality assurance in project planning, implementation, and deployment is required as defined in LIR 308-00-05.

Category 2 Software Project or product:

A Category 2 software project or product carries moderate to minimal impact in any of the identified assessment areas and is not Category 1 software. Accordingly, a simplified approach to SQA is allowed as defined in LIR 308-00-05.This category allows the tailoring of required processes to be commensurate with the intended use of the software.

Category 3 Software Project or product:

A Category 3 software project or product carries minimal to negligible impact in all of the identified assessment areas. Accordingly, LIR 308-00-05. requires an approach involving only the registration of the project and this risk assessment worksheet with the software developer's group and/or program office, as applicable.

Control:

A method or means used to mitigate hazards and reduce risk.

Hazard:

Any source or situation with potential to cause injury or harm to workers or the public, harm to the environment, incurred liability, or damage to or loss of property.

Initial risk:

The risk before controls are established and in place, or the risk posed by hazards that have not been adequately controlled; for activities with established control systems, the risk before Laboratory-instituted controls are implemented, or the risk posed because of the likelihood of failure of the established controls.

Risk:

A function of the likelihood and potential severity of injury, harm, incurred liability, damage or loss; a qualitative judgment based on knowledge and experience.

Software:

A computer program or suite of programs, procedures or routines, having material value and usefulness, that enables a computer system to process data.

- *Material value* - Automates Laboratory operations and/or is delivered for use or evaluation by one or more Laboratory customers.
- *Usefulness* - Is actively used today.
- *Enables* - Provides tools for and allows access to.
- *Process* - Performing data entry, data storage, data retrieval, calculations, or any other data manipulation.
- *Data* - Information organized for analysis or computation.

Worker:

Any Laboratory employee, any contract employee, subcontract employee or visitor engaged in performing work at the Laboratory.

Probable:

Likely to occur, or reasonably expected.

Possible:

Slight chance of occurrence, or highly unlikely to occur.

Not Credible:

Effectively zero probability of occurrence.

Attachment C
Institutional Software Quality Management Implementation Plan

1.0 Purpose

This implementation plan is associated with the Software Quality Management (SQM) LIR (308-00-05) and describes how and when the Laboratory will implement the LIR.

2.0 Scope

This implementation plan focuses primarily on the development of division-level implementation plans, which will describe individual division approaches to developing and implementing software quality management.

This implementation plan describes the institutional implementation approach, division-level implementation plan timeline requirements, institutional resources that are necessary to assist Laboratory divisions in developing their implementation plans and processes required by the LIR, risk analysis, and metrics that will be gathered to track progress in the development of the individual division implementation plans.

3.0 Risk Analysis

3.1 Management Support

Risk: Without institutional adoption and furtherance of the efforts of the SQA office and SESC, there is a substantial risk that SQM will not be implemented successfully at the Laboratory.

Mitigation: CIO and LANL management need to keep software quality management in the forefront of their decision process. Sufficient guidance, policies and resources shall be given to allow for the implementation of SQM. No software project should be approved without being graded appropriately. Each project should be evaluated for adherence to division policies on software management.

3.2 Experienced and Available Staff

Risk: If the SESC and divisions are not staffed with personnel that have sufficient experience in real-world software engineering practices, the effectiveness and timeliness of SQM implementation at the Laboratory will be negatively impacted.

Mitigation: The Laboratory must provide professional experienced software engineering personnel who have academic and real-world experience.

3.3 Credibility and Added Value

Risk: If the SESC and divisions develop inflexible, restrictive guidance and example products that do not fit the real-world situations of the Laboratory, division-level implementations of the LIR will be correspondingly restricted.

Mitigation: The CIO's office and the SESC must establish and maintain a reputation as competent, knowledgeable professionals in place to assist Laboratory staff with ISQM efforts. To accomplish this, the SESC must develop and release products that are timely, truly in demand, and must communicate to interested Laboratory staff the availability of current products and the progress being made on planned products. The Division must develop flexible, nonrestrictive guidance, processes and example products that fit the real-world situations of the Laboratory.

Guidance Note: A cultural change process involves coordination of management initiatives, staff processes, tools, and staff operations. It requires continuous communication about current efforts and feedback on past efforts, present performance, and future plans. Without patience and a continuous communication and feedback process, the SQM initiative will encounter too much resistance to succeed.

4.0 Institutional SQM Implementation Requirements

Software that falls within the scope of the LIR has great variety in type, scope and customer. Therefore, Laboratory SQM implementation efforts must be tailored and planned at the division level.

The Laboratory Chief Information Officer (CIO) shall continue to support the Software Engineering Process Group (SEPG) and shall establish a Software Quality Assurance (SQA) office and a Software Engineering Services Center (SESC). These resources shall provide guidance and software engineering subject matter experts who shall aid in the implementation of the LIR.

Operating with this assumption, this implementation plan shall extend only for four months after release of the LIR. The resources described above shall be available upon release of the SQM LIR, and this implementation plan shall be updated to support ongoing division implementation efforts to respond to the LIR.

Establishing the SQA office and SESC described above and securing funding for FY 04 shall be the responsibility of the Laboratory CIO and the Performance Surety Division Leader. The CIO must report regularly to the Laboratory Director and Senior Executive Team on the progress of SQM implementation.

5.0 Division-Level Implementation Requirements

Guidance Note: Although representatives of numerous divisions participated in the development of the LIR and several divisions already have some form of software quality management in place, it is recognized that many divisions will need a startup time and help to plan for implementation of the LIR. For this reason, guidance has been developed that covers division-level implementation planning. This guidance is intended to provide planning ideas and is not to be construed as Laboratory requirements.

Each Laboratory Division or Office (referred hereafter as "division") that is developing software as defined in the LIR shall review the SQM LIR and shall develop an implementation plan describing the approach, tasks and schedule it will follow to implement the LIR. The purpose of the division implementation plan shall be to describe how and when the division shall fulfill the requirements of the LIR, i.e., how the division shall ensure that software is developed using SQM processes appropriate to the risk category.

The division implementation plan must address the following:

- Roles and responsibilities for division SQM.
- Approach, tasks and schedule for how the division shall grade its software.
- Approach, tasks and schedule that shall be implemented for prioritizing and developing the software processes required by the LIR.
- Implementation approach, tasks and schedule for the divisions Category 1 and Category 2 software projects' execution of the processes required by the LIR.
- The approval process for adoption of the implementation plan, which must at a minimum provide the following information to the CIO's office:
    - Date the draft SQM implementation plan is complete.
    - Date the CIO SQA office reviews the draft implementation plan for conformance to LIR requirements.
    - Date the division leader approves the reviewed SQM implementation plan.
    - Date the cognizant associate director approves the SQM implementation plan.

The timeframe for developing the division implementation plan shall be four months from the release of the LIR for concurrence by the CIO, and one additional month for sign-off by the cognizant associate director.

Guidance Note: The following guidance is available to divisions beginning an implementation plan:

- Web site: Institutional Software Quality Management Program: http://isqm.lanl.gov
- Web site: DOE CIO: http://cio.doe.gov/ITReform/sqse/index.html

6.0 Institutional Metrics

The following data must be gathered from divisions regarding the development of their SQM implementation plans:

- Date the draft SQM implementation plan is complete.
- Date the CIO SQA office reviews the draft SQM implementation plan for conformance to LIR requirements.
- Date the division leader approves the reviewed SQM implementation plan.
- Date the cognizant associate director approves the SQM implementation plan.

This data shall be used to develop metrics describing the progress of development of division-level SQM implementation plans. The Laboratory goal shall be 100% development of division implementation plans within five months (four months concurrence by the CIO and one month sign-off by the cognizant associate director) following the release of the SQM LIR.